IN THE CLAIMS

1. (Original) A method of implementing random early detection of congestion for a data packet queue, the method comprising:

   storing at least one configurable segment-based drop probability profile associated with the queue, each segment of the profile having a drop probability associated with a range of average queue sizes;

   receiving an average queue size AQS for the queue;

   determining whether the average queue size AQS falls within one of N segments S_i, $0 \le i < N$, corresponding to the drop probability profile associated with the queue; and

   basing a drop/no-drop decision, for a data packet associated with the queue, on the drop probability associated with a segment S_i when AQS falls within that segment.

2. (Original) The method of claim 1, wherein each data packet is assigned a drop priority from among M possible drop priorities, $M > 1$, and wherein M drop probability profiles are associated with the queue, further comprising identifying, based on the drop priority of the data packet awaiting the drop/no-drop decision, which of the M drop probability profiles to use in determining whether the average queue size AQS falls within one of N segments S_i.

3. (Currently amended) The method of claim 2, wherein each of the M profiles comprises a high threshold T_H and a delta threshold T_δ, wherein T_δ is used to determine multiple segment boundaries, each boundary lower than T_H.

4. (Currently amended) The method of claim 1, wherein the configurable drop probability profile comprises a high threshold T_H and a delta threshold T_δ, wherein T_δ is used to determine multiple segment boundaries, each boundary lower than T_H.

5. (Original) The method of claim 4, wherein AQS falls within S_i when

$$\begin{array}{ll} T_H - T_\delta / 2^i \le AQS < T_H - T_\delta / 2^{i+1} & i \ne N-1 \\ T_H - T_\delta / 2^i \le AQS < T_H & i = N-1 \end{array}.$$

6. (Original) The method of claim 1, wherein the segment endpoints for each segment S_i are stored with the profile.

7. (Original) The method of claim 1, wherein a set of drop probabilities P_i correspond respectively to the segments S_i, wherein basing a drop/no-drop decision comprises, when *AQS* falls within S_i, performing a random trial having two outcomes, *drop* and *admit*, where the probability of the outcome *drop* is approximately P_i.

8. (Original) The method of claim 1, wherein multiple drop probability profiles exist, and wherein one set of drop probabilities P_i correspond to all profiles.

9. (Original) The method of claim 1, wherein multiple drop probability profiles exist, and wherein a set of drop probabilities P_i is stored with each profile.

10. (Original) The method of claim 1, wherein multiple drop probability profiles exist, and wherein each profile further comprises a segment type identifier, the segment type identifier corresponding to one of a plurality of fixed segment distributions.

11. (Currently amended) A method of implementing random early detection of congestion for multiple data packet queues sharing a common queue memory, the method comprising:

maintaining a pool of drop probability profiles, the number of profiles in the pool less than the number of queues;

associating each of the data packet queues with at least one drop probability profile selected from the pool;

maintaining a queue size for each data packet queue; and

forming a drop/no-drop decision for a data packet prior to queuing that data packet in its destination data packet queue, the decision based on the destination data packet queue size and the profile associated with that queue. ~~and~~

~~when a drop/no-drop decision is to be made for a data packet destined for a particular queue, using the profile associated with that queue to arrive at the drop/no-drop decision.~~

12. (Currently amended) The method of claim 11, wherein each profile in the profile pool comprises a high threshold T_H and a delta threshold T_δ, wherein T_δ is used to determine multiple segment boundaries, each boundary lower than T_H.

13. (Currently amended) The method of claim 11, wherein ~~using the profile associated with that queue~~ forming a drop/no-drop decision comprises determining whether ~~the~~ an average queue size *AQS* for that queue falls within one of *N* drop probability profile segments S_i, $0 \leq i < N$.

14. (Original) The method of claim 13, wherein *AQS* falls within S_i when

$$\begin{array}{ll} T_H - T_\delta / 2^i \leq AQS < T_H - T_\delta / 2^{i+1} & i \neq N-1 \\ T_H - T_\delta / 2^i \leq AQS < T_H & i = N-1 \end{array}.$$

15. (Original) The method of claim 13, wherein the segment endpoints for each segment S_i are stored with the profile.

16. (Original) The method of claim 13, wherein a set of drop probabilities P_i correspond respectively to the segments S_i, wherein using the profile associated with a queue further comprises, when *AQS* falls within S_i, performing a random trial having two outcomes, *drop* and *admit*, where the probability of the outcome *drop* is approximately P_i.

17. (Original) The method of claim 16, wherein one set of drop probabilities P_i correspond to all profiles.

18. (Original) The method of claim 13, wherein a set of drop probabilities P_i is stored with each profile.

19. (Original) The method of claim 13, wherein each profile further comprises a segment type identifier, the segment type identifier corresponding to one of a plurality of fixed segment distributions.

20. (Original) The method of claim 11, wherein each data packet is assigned a drop priority from among *M* possible drop priorities, and wherein each data packet queue is associated

with *M* drop probability profiles, wherein using the profile associated with that queue to arrive at the drop/no-drop decision comprises using the profile associated with that queue and the drop priority of that data packet.

21. (Original) An integrated circuit comprising:

a profile register to store an array of segment-based drop probability profiles;

a profile matcher to accept a data packet queue identifier and select an active drop probability profile, from the array of drop probability profiles, that is associated with the data packet queue identifier;

a segment selector to accept an average queue size *AQS* associated with the selected data packet queue and the active drop probability profile, the selector identifying the segment of the active profile within which *AQS* falls; and

a probability comparator to accept a drop probability corresponding to the segment identification from the segment selector, the comparator producing a random packet-drop signal with a probability approximately equal to the drop probability.

22. (Currently amended) The circuit of claim 21, wherein the drop probability profile configuration of the profile register comprises a high threshold T_H and a delta threshold T_δ for each entry in the array, wherein T_δ is used to determine multiple segment boundaries, each boundary lower than T_H.

23. (Original) The circuit of claim 22, wherein the segment selector has the capability to identify the segment as one of N segments S_i, $0 \le i < N$, wherein the identified segment S_i is the segment for which

$$\begin{aligned} T_H - T_\delta / 2^i \le AQS < T_H - T_\delta / 2^{i+1} &\quad i \ne N-1 \\ T_H - T_\delta / 2^i \le AQS < T_H &\quad i = N-1 \end{aligned}$$

24. (Original) The circuit of claim 22, wherein the drop probability profile configuration of the profile register further comprises a set of drop probabilities, corresponding respectively to the possible segment identification values produced by the segment selector.

25. (Original) The circuit of claim 22, wherein the drop probability profile configuration of the profile register further comprises a set of segment endpoints.

26. (Original) The circuit of claim 22, wherein the drop probability profile configuration of the profile register further comprises a segment type identifier, the segment type identifier corresponding to one of a plurality of fixed segment types defined for the circuit.

27. (Original) The circuit of claim 21, wherein the profile matcher comprises a profile pointer register having at least one entry corresponding to each valid data packet queue identifier, the profile matcher selecting an active drop probability profile by returning a pointer associated with the data packet queue identifier.

28. (Original) The circuit of claim 27, wherein the profile matcher further accepts a drop priority identifier, and wherein, for each valid data packet queue identifier, the profile pointer register has one entry corresponding to each valid drop priority identifier, the profile matcher selecting an active drop probability profile b returning the register entry corresponding to the data packet queue identifier and the drop priority identifier.

29. (Original) The circuit of claim 21, further comprising:

an average queue size register to store an array of average queue size elements AQS_i, one element corresponding to each valid data packet queue identifier, and to supply the array element corresponding to the data packet queue identifier to the segment selector as AQS; and

an average queue size filter to update an average queue size element AQS_i with a current queue size CQS_i, where both AQS_i and CQS_i are associated with a selected data packet queue.

30. (Original) The circuit of claim 29, wherein the filter operates according to the equation

$$AQS_i = AQS_i + \alpha(CQS_i - AQS_i),$$

where $0 < \alpha < 1$.

31. (Original) The circuit of claim 30, wherein $\alpha = 2^{-n}$, where n is a positive integer.

32. (Original) The circuit of claim 21, further comprising a random number generator to generate random numbers for use by the probability comparator.

33. (Original) A packet routing device comprising:

a packet pipeline to receive and delay a packet;

a multiple-queue packet memory to receive a packet into the queue designated for that packet, when that packet reaches the end of the packet pipeline without being dropped; and

a random early detection traffic conditioning circuit to snoop queue information from the packet while the packet is in the pipeline, the circuit having the capability to select a segment-based drop probability profile corresponding to the queue information and produce a packet drop/no-drop decision based on that profile before the packet reaches the end of the pipeline.

34. (Original) The device of claim 33, wherein the packet pipeline has the capability to maintain the current queue size for each queue in the multiple-queue packet memory, and wherein the random early detection circuit uses the current queue size from the packet pipeline to maintain an average queue size for each queue in the multiple-queue packet memory.

35. (Original) The device of claim 33, wherein the queue information corresponds to a destination port and a traffic class for the packet.

36. (Original) The device of claim 33, wherein the random early detection circuit accepts a drop priority for the packet, and where the selection of a drop probability profile depends on the drop priority as well as on the destination port and traffic class of the packet.

37. (Original) The device of claim 36, further comprising a traffic monitor circuit to produce the drop priority for the packet while the packet is in the packet pipeline.

38. (Original) The device of claim 33, wherein the random early detection circuit comprises a bank of user-configurable drop probability profiles and a user-configurable profile pointer register having at least one pointer, pointing to a drop probability profile from the bank, for each queue in the multiple-queue packet memory, the random early detection

circuit using the queue information to select a profile pointer from the register, thereby selecting the drop probability profile corresponding to the queue information.

39. (Original) The device of claim 33, further comprising multiple packet ingress interfaces, each ingress interface coupled to supply packets to the packet pipeline.

40. (Original) An apparatus comprising a computer-readable medium containing computer instructions that, when executed, cause a processor or multiple communicating processors to perform a method for random early detection of congestion in a data packet queue, the method comprising:

storing at least one configurable segment-based drop probability profile associated with the data packet queue, each segment of the profile having a drop probability associated with a range of average queue sizes;

receiving an average queue size *AQS* for the queue;

determining whether the average queue size *AQS* falls within one of *N* segments S_i, $0 \leq i < N$, corresponding to the drop probability profile associated with the queue; and

basing a drop/no-drop decision, for a data packet associated with the queue, on the drop probability associated with a segment S_i when *AQS* falls within that segment.

41. (Original) The apparatus of claim 40, wherein each data packet is assigned a drop priority from among *M* possible drop priorities, and wherein *M* drop probability profiles are associated with the queue, further comprising identifying, based on the drop priority of the data packet awaiting the drop/no-drop decision, which of the *M* drop probability profiles to use in determining whether the average queue size *AQS* falls within one of *N* segments S_i.

42. (Currently amended) An apparatus comprising a computer-readable medium containing computer instructions that, when executed, cause a processor or multiple communicating processors to perform a method for random early detection of congestion in multiple data packet queues sharing a common queue memory, the method comprising:

maintaining a pool of drop probability profiles, the number of profiles in the pool less than the number of queues;

dynamically associating each of the data packet queues with at least one drop probability profile selected from the pool;

maintaining a queue size for each data packet queue; and

forming a drop/no-drop decision for a data packet prior to queuing that data packet in its destination data packet queue, the decision based on the destination data packet queue size and the profile associated with that queue. ~~and~~

~~when a drop/no-drop decision is to be made for a data packet destined for a particular queue, using the profile associated with that queue to arrive at the drop/no-drop decision.~~